Supplementary Document for Paper 1227

Category: Research

1 Introduction

We provide additional details on prior methods, more results and comparisons, discuss implementation details and provide statistical analysis of the immersive user study.

2 Prior 2-step Decomposition Methods

We compare our 2D velocity computation algorithm BAM (Section 4 in main text) to the following prior methods:

2.1 Universal Power Law

This is a social force (SF) based model that treats the crowd as a collection of mass particles [1]. Newtonian-like physics is applied to the system to compute the trajectories of the agents. At each time step, the superposition of various forces are computed for each agent, ultimately determining a feasible velocity by imparting acceleration on the agent. The preferred velocity is converted into a driving force. The agent avoids collisions with other agents and obstacles through the application of repulsive forces. In contrast to prior social-forces based methods, the authors analyze large databases of captured pedestrian trajectories and find that the interaction energy between two pedestrians is inversely proportional to the square of the projected time to a potential future collision. The interaction energy is then used to derive the repulsive force, and overall acceleration. Our implementation is based on the publicly available source code, available at http://motion.cs.umn.edu/PowerLaw/, and the original values given by the authors.

2.2 ORCA

Optimal Reciprocal Collision Avoidance (ORCA) [3] applies geometric optimization techniques in velocity space. It is based on the notion of velocity obstacles from robotics. Avoiding collision simply requires selecting a velocity which does not lie within the reciprocal velocity obstacle set for each agent. ORCAs unique formulation defines the velocity obstacles as half planes. The set of velocities that are permitted for agent with respect to all other agents is the intersection of the half planes of permitted velocities induced by each other agent. It uses 2D linear programming to efficiently find a velocity in the set of permitted velocities which minimizes the distance to the preferred velocity. We use the publicly available source code, available athttp://gamma.cs.unc.edu/ORCA/, and the original values given by the authors.

3 Additional Results

We evaluate our 2D navigation algorithm, BAM, with prior methods based on velocity obstacles (ORCA) and social forces (Powerlaw). In each case, we couple the 2D planner with the same animation system which is capable of precisely following a 2D trajectory. The BAM algorithm accounts for the full body pose during 2D planning leading to fewer collisions in the synthesized motion (Table 1 in the main document). Moreover, it generates smoother trajectories at a slightly higher run time cost. Provided here are details of the evaluation criterion and a discussion of the results.

3.1 Agent-Agent Collision Rate Metric

We estimate the rate of agent-agent collisions by analyzing the 2D position of the bounding disc of the underlying articulated character. Our coupled approach ensures that the position of the disc is synchronized with that of the root joint of the corresponding skeletal mesh (Section 3.4 of the main document). Penetration Depth is a common metric used for quantifying collisions. It can be defined as the minimum displacement required to eliminate overlap between two entities. For each pair of adjacent time steps, we compute the maximum penetration depth (PD_{ij}) between two agents, i and j, over the interval bound by those time steps t^k and t^{k+1}. Assuming that agents move linearly between time steps, the position of agent i during the interval $[t^k, t^{k+1}]$ can be given as $\mathbf{p}_i(t) = \mathbf{p}_i + \mathbf{v}_i t, t^k \leq t \leq t^{k+1}$, where $\mathbf{p}_i$ and $\mathbf{v}_i$ denote the position and velocity respectively of agent i at time t = t^k. The maximum penetration depth can be computed by finding the minimum distance, or, equivalently, the minimum squared distance, between the two agents, i and j, over the time interval $[t^k, t^{k+1}]$ as:

$$d_{ij}^{min} = \min_{0 \leq s \leq 1} \|(\mathbf{p}_i + \mathbf{v}_i^{'} s) - (\mathbf{p}_j + \mathbf{v}_j^{'} s)\|^2, \quad (1)$$

where $\mathbf{v}_i^{'} = (t^{k+1} - t^k)\mathbf{v}_i$ and $s = -\frac{\mathbf{p}_{ij} . \mathbf{v}_{ij}^{'}}{\|\mathbf{p}_{ij}\|^2}$ for relative position $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ and relative velocity $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. We can normalize the maximum penetration depth over the time step as $PD_{ij}^k = max(0, 1 - d_{ij}^{min}/r_{ij})$ where r_{ij} denotes the sum of the radii of the two discs. Finally, the collision rate C for the simulation can be computed by averaging all frames and agents:

$$C = \frac{1}{TN} \sum_{t=0}^{N} \sum_{i=0}^{N} \sum_{j=i+1}^{N} PD_{ij}^t, \quad (2)$$

where N is the number of agents, and T is the number of time steps.

Table 1 in the main document presents the collision rates of the three methods on different benchmarks (Section 7.2.1 of the main document). The BAM algorithm accounts for the current pose of the skeletal mesh and several human motion constraints to generate velocities that are amenable to motion synthesis. This two-way coupling reduces the mismatch between the 2D planning and full body synthesis, and reduces collisions.

3.2 Trajectory Smoothness Metric

Similar to the above section, we analyze the 2D trajectory of the root joint. Smaller accelerations are likely to generate smoother motions. The smoothness score A is simply the average acceleration over all the agents and all the simulation steps:

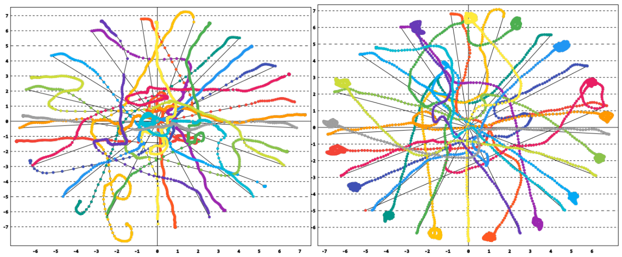
$$A = \frac{1}{TN} \sum_{t=0}^{N} \sum_{i=0}^{N} \|\dot{\mathbf{v}}_i^t\|, \quad (3)$$

where N is the number of agents, T is the number of time steps, and $\dot{\mathbf{v}}_i^t$ denotes the acceleration of agent i at step t.

In all benchmarks, BAM generates smoother trajectories compared to ORCA and SF, as indicated by the relatively low average acceleration scores listed in Table 1 in the main document.

3.3 Comparison with Prior Coupled Approaches

We evaluated the smoothness of the trajectories generated by our algorithm, with those generated by Smartbody on the anti-podal circle benchmark with 17 agents. We plot these trajectories in Figure 1. The agents in our approach are able to navigate to their goals faster with smoother trajectories. On the other hand, Smartbody



(A) SmartBody [2] (B) BAM (Our Algorithm)

Figure 1: **Trajectory Comparisons on the Antipodal circle benchmark.** For each simulation, we visualize the root joint position of each agent using a different color. (a) The agent trajectories generated in Smartbody (using Steerlib) exhibit jittering and several collisions, as indicated by the high density of trajectories at the center. (b) Our method, BAM, results in fewer collisions and smoother trajectories. Moreover, our agents avoid the congested center and reach their goals faster than Smartbody.

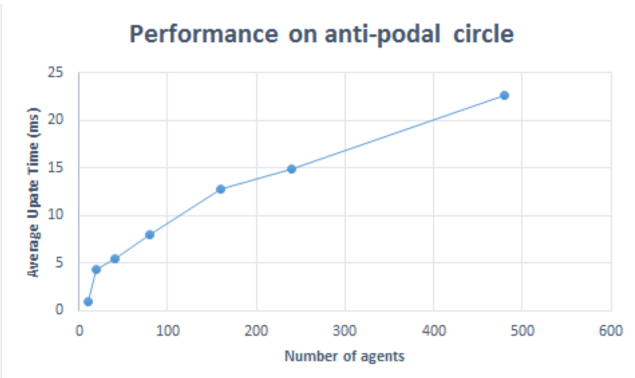


Figure 2: **Performance Graph** This graph shows the performance of our algorithm BAM on the anti-podal circle with increasing numbers of agents. BAM can compute collision-free trajectories of hundreds of agents at interactive rates while accounting for kinematic and dynamic stability motion constraints. Our approach can also simulate 60+ agents at 60 fps, including full body motion synthesis and rendering.

can generate noisy trajectories with significantly higher number of collisions. This is denoted by the jittery trajectories in the middle region near the center of the circle in Figure 1(b). We also present results on other benchmarks in Table 1 in the main text.

4 Implementation & Performance

We have implemented our algorithm in C++ on a Windows 10 desktop PC. All the timing results in the paper were generated on an Intel Xeon E5-1620 v3 with 4 cores and 16 GB of memory. We present the timing results (Figure 2) on the anti-podal circle benchmark where agents are placed on the circumference of the circle with diametrically opposite goals. BAM offers comparable performance to ORCA and can simulate 100s of 2D agents at interactive rates. Our overall approach couples BAM with a motion synthesis system. We can simulate 60+ full-body agents at 60+ fps which includes BAM, motion synthesis and rendering costs.

4.1 Full-body Motion Synthesis

We couple our novel 2D velocity computation algorithm (BAM) with a relatively simple animation system. The animation system uses linear blending between three motion clips. The motion clips represent the same motion, just played back at different speeds. This type of synthesis is physically incorrect since. it does not capture the relationship between walking speed and stride length. It can also result in footskating artefacts. Overall, the animation system is fairly simple but can precisely follow a synthesized 2D trajectory. When comparing against prior 2D navigation methods, we consistently use the same animation system. Despite the abovementioned limitations in our full-body animation system, we find that BAM provides quantitative and qualitative benefits over prior methods. BAM can be easily integrated with a number of existing animation systems, each of which may offer its own unique advantages and disadvantages.

4.2 Parallelization potential

As mentioned in Section 5.4 of the main text, our algorithm is currently not optimized and can benefit from parallelization. We use an agent-based velocity computation algorithm, which relies only on the current observable state of other agents (current position, velocity, orientation) and local environment information to independently compute the new velocity for each agent. Thus, at every timestep we can sync the simulation and then each agent can independently compute new velocities in parallel and cache these velocities in a temporary variable. Once all agents have computed their new velocities, we can update the current velocity of each agent to be equal to its recently computed new velocity.

5 Agent-Agent & Avatar-Agent Interactions in Immersive Settings

As described in Section 6.2 of the main text, the immersive study comprised of two conditions: `avatar-agent condition` where the user by means of his/her avatar can freely move around and interact with the agents; and `avatar-agent condition` where the user observes agent-agent interactions from a distance. In both conditions the user embodies his/her avatar with a first person perspective rendering through the HTC Vive HMD. We now describe the specific questions, the statistical response, and analysis for the two questions.

5.1 Active Avatar-Agent Interactions

In this condition, the avatar can freely interact with the agents as he/she engages in the task of moving boxes. We provide below, the specific questions and the responses of the user. It can be seen that responses were statistically significant and in favour of BAM compared to both ORCA and powerlaw on the questions of "In which simulation did you feel the virtual characters were more aware of your presence?" and "In which simulation were the virtual characters avoiding you more?". We plot the frequency of user's responses for these questions in Figure 4 and Figure 3 respectively. Each case denotes a high frequency of strong responses for BAM i.e. a response of 1.

Statistical Results Per Question: The following are the questions asked in this condition. Each questionnaire compared our method to one of either Powerlaw or ORCA. A rating of 1 indicates a strong preference for BAM, 7 indicates a strong preference for the other method and 4 indicates no preference. We provide the mean and std deviation for each question. Moreover, questions for which responses were statistically significant ($p < 0.001$) are in bold.

- In which simulation did you have a greater sense of being in the same space as the virtual characters?
  - **ORCA:** 3.6 ± 1.53
  - PowerLaw 3.44 ± 1.79
- In which simulation did you respond more to the virtual characters?
  - ORCA: 3.88 ± 1.82
  - PowerLaw 3.88 ± 1.78

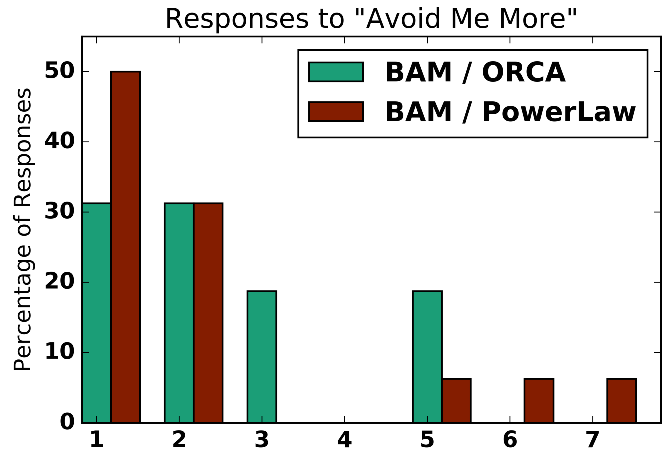


Figure 3: **Frequency of user responses for avatar-agent interactions on the question of avoidance.**: Participants responses to the question "In which simulation were the virtual characters avoiding you more?". A rating of 1 indicates a strong preference for BAM, 7 indicates a strong preference for the other method and 4 indicates no preference. Participants felt that BAM agents avoided them significantly more than Powerlaw or ORCA.

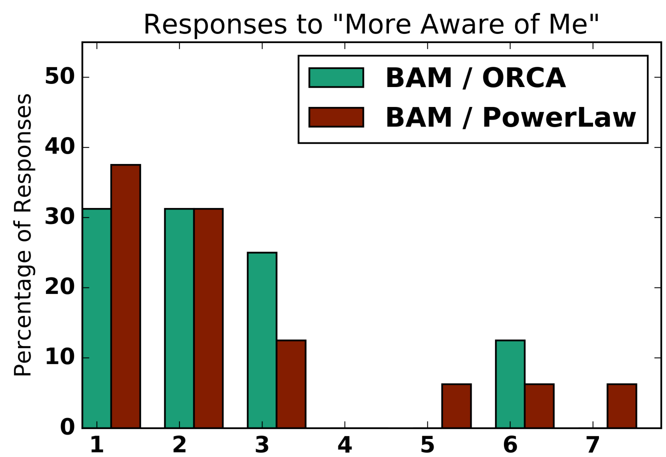


Figure 4: **Frequency of user responses for avatar-agent interactions on the question of awareness.** Participants responded to the question "In which simulation did you feel the virtual characters were more aware of your presence?" A rating of 1 indicates a strong preference for BAM, 7 indicates a strong preference for the other method and 4 indicates no preference. Participants felt that BAM agents were more aware of them compared to prior methods. This indicates that participants felt more co-presence with BAM agents.

- In which simulation did you feel the virtual characters were more aware of your presence?
  - **ORCA:** 2.44 ± 1.59
  - **PowerLaw** 2.50 ± 1.90
- In which simulation were the virtual characters avoiding you more?
  - **ORCA:** 2.25 ± 1.95
  - **PowerLaw** 2.44 ± 1.46
- In which simulation were the virtual characters avoiding you more naturally?
  - ORCA: 2.88 ± 1.93
  - PowerLaw 3.31 ± 2.24
- In which simulation did the virtual characters follow more natural paths?
  - ORCA: 3.31 ± 1.66
  - PowerLaw 4.0 ± 2.07
- In which simulation did the presence of the virtual characters affect you more in the way you explored the space?
  - ORCA: 4.25 ± 2.21
  - PowerLaw 4.56 ± 1.97

Discussion:

It can be clearly seen that BAM outperformed Powerlaw and ORCA in all cases where the responses were statistically significant. The average responses were in favour of BAM but are not reliable measures due to lack of significance. Our user study comprised of 18 subjects out of which data for 2 subjects was discarded. Thus, we analyzed data for only 16 subjects which can potentially explain the lack of signiciance in some of the questions. For our analysis in Section 6.2 of the main text, we limit ourselves to two questions for which we had statistically significant responses in both comparisons. The frequency of user responses for these questions is also provided in Figure 4 and Figure 3 respectively.

5.2 Passive Observations of Agent-Agent Interactions

In this condition, the user by means of his/her avatar simply observes multi-agent interactions from a distance. The following are the questions asked in this condition. Each questionnaire compared our method to one of either Powerlaw or ORCA. A rating of 1 indicates a strong preference for BAM, 7 indicates a strong preference for the other method and 4 indicates no preference. We provide the mean and std deviation for each question. Moreover, questions for which responses were statistically significant ($p < 0.001$) are in bold.

- In which simulation did the virtual characters exhibit fewer collisions?
  - ORCA: 3.944 ± 1.88
  - **PowerLaw** 2.88 ± 1.50
- In which simulation did the virtual characters exhibit fewer artefacts? [Artefacts include abnormal actions such as spinnning in place, unnatural foot placement, foot 'skating', etc]
  - ORCA: 4.69 ± 1.45
  - PowerLaw 3.56 ± 1.46
- In which simulation did the virtual characters exhibit more natural interactions?
  - ORCA: 4.0 ± 1.63
  - PowerLaw 3.81 ± 1.64
- In which simulation did the virtual characters follow more natural paths?
  - ORCA: 4.06 ± 1.69
  - PowerLaw 3.5 ± 1.71

None of the questions had statistically significant responses for both methods. Hence, this data is hard to analyze and requires more evaluation.

References

[1] I. Karamouzas, B. Skinner, and S. J. Guy. Universal power law governing pedestrian interactions. *Physical Review Letters*, 113(23):238701, 2014.
[2] A. Shapiro. Building a character animation system. In J. Allbeck and P. Faloutsos, eds., *Motion in Games*, vol. 7060 of *Lecture Notes in Computer Science*, pp. 98–109, 2011.
[3] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Inter. Symp. on Robotics Research*, pp. 3–19, 2011.